

This is the peer reviewed version of the following article:

Zenoni G., Leicht T., Colombo A., Botti L.

An agglomeration-based adaptive discontinuous Galerkin method for compressible flows
International Journal for Numerical Methods in Fluids. 2017;0:1–19,

which has been published in final form at <https://doi.org/10.1002/fld.4390>.

This article may be used for non-commercial purposes in accordance with Wiley Terms and Conditions for Self-Archiving.

An agglomeration-based adaptive Discontinuous Galerkin method for compressible flows

G. Zenoni¹, T. Leicht², A. Colombo^{1*}, L. Botti¹

¹*Università degli Studi di Bergamo, Dipartimento di Ingegneria e Scienze Applicate, Italy*

²*Institut für Aerodynamik und Strömungstechnik, Deutsches Zentrum für Luft- und Raumfahrt e.V. (DLR), Germany*

SUMMARY

In this work we exploit the possibility to devise Discontinuous Galerkin discretizations over polytopic grids to perform grid adaptation strategies based on agglomeration coarsening of a fine grid obtained via standard unstructured mesh generators. The adaptive agglomeration process is here driven by an adjoint-based error estimator. We investigate several strategies for converting the error field estimated solving the adjoint problem into an agglomeration factor field, that is an indication of the number of elements of the fine grid that should be clustered together to form an agglomerated element. As a result the size of agglomerated elements is optimized for the achievement of the best accuracy for given grid size with respect to the target quantities. To demonstrate the potential of this strategy we consider problem-specific outputs of interest typical of aerodynamics, *e.g.* the lift and drag coefficients in the context of inviscid and viscous flows test cases. Copyright © 2017 John Wiley & Sons, Ltd.

Received ...

KEY WORDS: Discontinuous Galerkin methods, compressible Euler and Navier–Stokes equations, adjoint problem, *h*-adaptivity, agglomerated meshes, aerodynamics

1. INTRODUCTION

In the last decades Computational Fluid Dynamics (CFD) has become a widespread practice in several industrial fields, *e.g.* aerodynamics, turbomachinery. The growing need of high-fidelity flow simulations for the accurate determination of problem-specific functionals of interest paved the way to higher-order methods such as the Discontinuous Galerkin (DG) method and sophisticated tools for the error estimation, *e.g.* adjoint-based estimators. In recent years, the capability of DG methods to produce accurate solutions for both inviscid and viscous flows on grids composed of elements of very general shape (polytopes with possibly curved sides) has been demonstrated, see [1, 2, 3].

The increased geometrical flexibility provided by DG discretizations lead the way for new grid adaptation strategies. For an extensive review on output-based mesh adaptation for CFD see the work of Fidkowski and Darmofal and the references cited therein [4]. Aside from grid adaptation strategies based on elements subdivision, see *e.g.* Hartmann and Leicht [5, 6], or conforming grid modification, see *e.g.* Dolejší and Felcman [7], an alternative approach, based on the adaptive coarsening of a fine mesh, has been proposed by Bassi *et al.* [8] in the context of DG methods and applied by Di Pietro and Specogna to Hybrid High-Order formulations of electrostatics [9]. Recently Collis and Houston also developed an agglomeration-based adaptive mesh refinement algorithm coupled with a goal-oriented error estimator applied to the DG approximation of the linear elasticity equations for a homogeneous isotropic material [10].

*Correspondence to: alessandro.colombo@unibg.it

The adaptive coarsening strategy is attractive from the implementation viewpoint as the background mesh does not change and no nodes nor elements are dynamically added, moved or removed. The fine background mesh can be obtained with standard mesh generators and can be designed to provide a sufficiently accurate representation of the domain boundaries without stringent requirements on the grid cardinality. Indeed the adaptation strategy does not require to solve on the background grid which simply provides a skeleton to construct the agglomerated elements grid and a backup to perform numerical integration. Agglomerated elements grids are obtained clustering together the elements of the standard grid, thereby obtaining a coarser grid composed of arbitrarily shaped polytopes. As a side note, it is clear that this strategy does not require CAD integration since the domain discretization is provided once and for all by the fine grid. This means that during the adaptation cycles the agglomeration process does not affect at all the discrete representation of the boundaries provided by the underlying mesh while the cardinality and density of the adapted mesh is driven by the error estimator. This feature is particularly appealing when using DG methods where the geometrical approximation of curved boundaries has to be accurate enough even on very coarse meshes so as not to impair the accuracy of the high-order solution approximation [11].

While in [8] the grid adaption capabilities were demonstrated on model problems, relying on *a priori* knowledge of the analytical solutions to estimate discretization errors, in this paper we consider inviscid and viscous flow problems and drive the adaptive agglomeration process by means of adjoint-based error estimators. The adjoint problem has been shown to be effective for the definition of *a posteriori* and goal-oriented error estimators that can be employed to drive mesh and polynomial degree adaptation strategies, see [5, 12, 13, 14]. For achieving adaptive coarsening by agglomeration the error field should be converted into an agglomeration factor field, that is an indication of the number of elements of the fine grid that should be clustered together to form an agglomerated element. Once the agglomeration field is computed, graph based mesh partitioning algorithms or geometric multigrid agglomerators can be employed to obtain the agglomerated elements mesh (in general geometric multigrid agglomerators are best suited because they do not focus on the achievement of a certain number of partitions). It is important to remark that the agglomerated grid is generated from scratch at each step of the adaptation algorithm based on the sole agglomeration factor information. The agglomerated elements' sizes are influenced by the agglomeration factor while their shapes are determined by the agglomeration algorithm at hand, which might be designed to optimize aspect ratios or minimize the number neighbors in the graph representation of the grid. Clearly the agglomeration factor field is the key to optimize the mesh step size towards the achievement of the best accuracy with respect to the target quantities.

In this work several techniques for coupling the adjoint-based error estimator with an agglomeration-based mesh adaptation strategy are investigated. The strategies share in common the following features: *i*) each element of the fine grid has its agglomeration factor indication, which determines the number of elements it should be clustered with; *ii*) the adaptation process is initialized with a uniform agglomeration factor which evolves according to the error field; *iii*) the elemental agglomeration factor value evolves according to the history of the adaption process. The effectiveness and flexibility of the resulting strategy is demonstrated in practice by considering aerodynamics computations. The accuracy in estimating the aerodynamic coefficients by means of the adaptive mesh refinement strategy is compared with the results obtained based on uniformly agglomerated grids.

The paper is organized as follows. In Sec. 2 we briefly introduce the governing equations and the numerical tools needed to deal with agglomerated meshes in the DG context. Sec. 3 deals with the adjoint problem in its *p*-enrichment version and the related error estimator, also considering multiple target quantities. In Sec. 4 we present and discuss in detail some strategies to translate the adjoint-based error estimation into an agglomeration factor field. In the same section numerical results for both inviscid and viscous flows, also using a multi-target error estimator approach, are reported. Finally, conclusions are given in Sec. 6.

2. BASICS CONCEPTS FOR DG ON AGGLOMERATED MESHES

In this section the main numerical tools needed to deal with arbitrarily shaped elements in the DG context are introduced.

2.1. Discrete DG space and physical frame basis function

Let the domain Ω be a bounded connected open subset of \mathbb{R}^d , $d = 2$, and \mathcal{T}_h denote a mesh of Ω . Even if this work focuses on a two-dimensional space extension to three space dimensions is trivial and effortless. The computational domain Ω_h , approximating Ω to some extent, is partitioned into a finite set of non-overlapping arbitrarily shaped elements $T \in \mathcal{T}_h$ such that

$$\bar{\Omega}_h = \bigcup_{T \in \mathcal{T}_h} \bar{T}. \quad (1)$$

For constructing spatial DG discretizations we rely upon the following broken polynomial spaces

$$\mathbb{P}_d^k(\mathcal{T}_h) \stackrel{\text{def}}{=} \{v_h \in L^2(\Omega) \mid v_h|_T \in \mathbb{P}_d^k(T), \forall T \in \mathcal{T}_h\}, \quad (2)$$

where k is a non-negative integer and $\mathbb{P}_d^k(T)$ is the restriction to T of the polynomials functions of d variables and global degree at most k .

We remark that polynomials are defined in physical coordinates, as proposed by *e.g.* [15, 16, 17, 18, 19, 20]. To build a satisfactory arbitrarily high-order basis for $\mathbb{P}_d^k(\mathcal{T}_h)$ we orthogonalize monomial basis functions by means of the modified Gram-Schmidt (MGS) algorithm [21]. To ensure that the orthogonalization process is not affected by round-off errors when dealing with elongated or stretched elements the initial set of monomials is defined in a local reference frame relocated in the barycenter and aligned with the principal axis of inertia of T , for each $T \in \mathcal{T}_h$, as proposed by Bassi *et al.* [8]. On the one hand broken polynomial spaces in (2) provide optimal approximation properties independently of reference to physical frame mappings, see [22] for details, on the other hand the orthogonalization procedure ensures that the mass matrix is unit diagonal, irrespectively of the elements shapes.

2.2. Agglomerated meshes

Consider a standard element mesh \mathcal{R} of the computational domain Ω , such that *i)* for each element $E \in \mathcal{R}$ there exist a reference polygon E^{ref} and a polynomial mapping $\Psi_E : E^{\text{ref}} \rightarrow E$ such that $E = \Psi_E(E^{\text{ref}})$; *ii)* quadrature rules of arbitrary order are available on each E^{ref} .

A general mesh \mathcal{T}_h can be obtained by agglomeration coarsening of \mathcal{R} . Accordingly, every agglomerated element $T \in \mathcal{T}_h$ is an open bounded connected subset of Ω_h and, for each $T \in \mathcal{T}_h$, there exists $\mathcal{R}_T \subset \mathcal{R}$ such that

$$\bar{T} = \bigcup_{E \in \mathcal{R}_T} \bar{E}, \quad (3)$$

see Figure 1. The number of agglomerated elements composing \mathcal{T}_h can be controlled by tuning the cardinality of each mesh subset \mathcal{R}_T . Moreover, \mathcal{T}_h and \mathcal{R} provide the same approximation of the computational domain, namely Ω_h .

In this work, exploiting the strategy introduced in [8], we drive the local mesh step size by specifying, for each mesh element $E \in \mathcal{R}$, the agglomeration factor a^E , that is an upper bound for the number of sub-elements that can be clustered together with E to form an agglomerated element T . As a result, for each $T \in \mathcal{T}_h$, we get

$$\text{card}(\mathcal{R}_T) \leq \min_{E \in \mathcal{R}_T} (a^E). \quad (4)$$

The final step for designing an h -adaptive algorithm is to properly translate the error metric provided by an error estimator to an indication for the agglomeration factor a^E , see Bassi *et al.* [8].

Nevertheless, it is important to remark that specifying a^E for each mesh element $E \in \mathcal{R}$ is not sufficient to uniquely determine the coarse grid \mathcal{T}_h . Indeed, several different agglomerated elements meshes (possibly with different grid cardinalities) might satisfy Eq. (4). Figure 2 shows one among the possible grids \mathcal{T}_h that can be obtained based on the agglomeration factor field associated to the fine grid \mathcal{R} . In this work, agglomerated elements grids are obtained relying on a modified version of the MGridGen library originally developed in the framework of geometric multigrid by Moulitsas and Karypis [23]. The agglomeration algorithm takes into account the agglomeration factor field but also optimizes the aspect ratios of agglomerated elements according to the heuristics included in the MGridGen library, see [24]. The CPU time needed for mesh agglomeration, performed only once and during pre-processing, is negligible with respect to operators assembly and solution time.

Once agglomerated elements are built, also their faces need to be identified in order to define trace operators and fluxes of the DG discretization. Faces of an agglomerated element $T \in \mathcal{T}_h$ are defined as a portion of ∂T such that either $F = \partial T \cap \partial\Omega$ or there exists $T' \in \mathcal{T}_h$, $T' \neq T$, such that $F = \partial T \cap \partial T'$. Facets of an element $E \in \mathcal{R}$ are defined as a portion of ∂E such that there exists a (hyperplanar) face σ^{ref} of the corresponding reference element E^{ref} such that σ is the image of σ^{ref} through the mapping Ψ_E . For every face F the set Σ_F collects the facets σ partitioning F , i.e.,

$$\bar{F} = \bigcup_{\sigma \in \Sigma_F} \bar{\sigma}.$$

Faces of agglomerated elements are regrouped in the set \mathcal{F}_h . \mathcal{F}_h is partitioned into the set \mathcal{F}_h^b , collecting the faces located on the boundary of the computational domain, and the set \mathcal{F}_h^i , collecting the internal faces located at inter-element boundaries, such that $\mathcal{F}_h = \mathcal{F}_h^i \cup \mathcal{F}_h^b$. For all $F \in \mathcal{F}_h^b$, \mathbf{n}_F denotes the unit outward normal to Ω_h , whereas, for all $F \in \mathcal{F}_h^i$, the unit normal is arbitrary but fixed, e.g. pointing from T to T' .

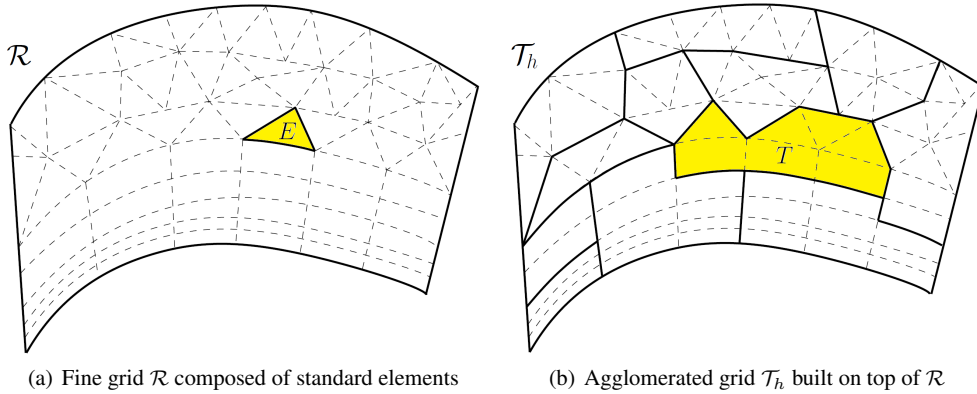


Figure 1. Example of a standard elements mesh \mathcal{R} , and an agglomerated elements mesh \mathcal{T}_h . Facets in \mathcal{T}_h are sketched with thick lines

2.3. Governing equations and DG discretization

In the two-dimensional case the Navier–Stokes equations can be written in compact conservation form as

$$\mathbf{P}(\mathbf{w}) \frac{\partial \mathbf{w}}{\partial t} + \nabla \cdot \mathbf{F}_c(\mathbf{w}) + \nabla \cdot \mathbf{F}_v(\mathbf{w}, \nabla \mathbf{w}) = \mathbf{0}, \quad (5)$$

equipped with suitable initial and boundary conditions, where \mathbf{w} is the vector of the m conservative variables, $\mathbf{F}_c, \mathbf{F}_v \in \mathbb{R}^m \otimes \mathbb{R}^d$ are defined as the convective and viscous flux functions, respectively, and d is the space dimension. In this work we rely on the set of the $m = 2 + d$ conservative variables $\mathbf{w}_c = [\rho, \rho u, \rho v, \rho e_0]^t$. With this choice the matrix $\mathbf{P}(\mathbf{w}) \in \mathbb{R}^m \otimes \mathbb{R}^m$ does not depend on \mathbf{w} and reduces to identity matrix $\mathbf{P} = \mathbf{I}$. Alternatives to the set of conservative variables for compressible flows were investigated by several authors, in these cases the transformation matrix $\mathbf{P}(\mathbf{w})$ can

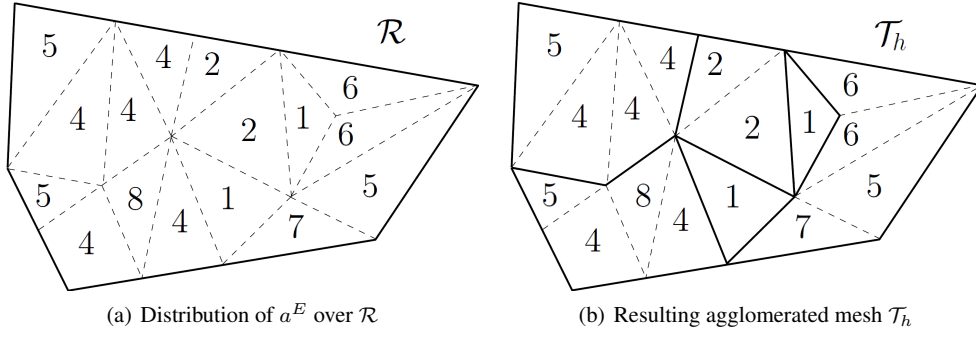


Figure 2. Examples of a background fine mesh \mathcal{R} with a^E distribution, and of a possible mesh \mathcal{T}_h resulting from the adaptive agglomeration process

differ from **I**, see [25, 26]. The Cartesian components $\mathbf{f}_c(\mathbf{w})$ and $\mathbf{g}_c(\mathbf{w})$ of the convective $\mathbf{F}_c(\mathbf{w})$ flux function, equal

$$\mathbf{f}_c(\mathbf{w}) = \begin{bmatrix} \rho u \\ \rho h_0 u \\ \rho u u + p \\ \rho v u \end{bmatrix}, \quad \mathbf{g}_c(\mathbf{w}) = \begin{bmatrix} \rho v \\ \rho h_0 v \\ \rho u v \\ \rho v v + p \end{bmatrix}, \quad (6)$$

where ρ is the fluid density, p is the pressure, u and v are the velocity components, e_0 and h_0 are the total energy and enthalpy per unit mass, respectively. The total enthalpy per unit mass is defined as $h_0 = e_0 + p/\rho$. By assuming that the fluid obeys to the perfect gas state equation, p can be computed as $p = (\gamma - 1) \rho [e_0 - (u^2 + v^2)/2]$ where γ indicates the ratio between the specific heats of the fluid. The Cartesian components $\mathbf{f}_v(\mathbf{w}, \nabla \mathbf{w})$ and $\mathbf{g}_v(\mathbf{w}, \nabla \mathbf{w})$ of the viscous flux function $\mathbf{F}_v(\mathbf{w}, \nabla \mathbf{w})$, are given by

$$\mathbf{f}_v(\mathbf{w}, \nabla \mathbf{w}) = -\mu \begin{bmatrix} 0 \\ u [2\partial_x u + \lambda (\partial_x u + \partial_y v)] + v (\partial_x v + \partial_y u) + (\gamma/\text{Pr}) \partial_x e \\ 2\partial_x u + \lambda (\partial_x u + \partial_y v) \\ \partial_x v + \partial_y u \end{bmatrix}, \quad (7)$$

$$\mathbf{g}_v(\mathbf{w}, \nabla \mathbf{w}) = -\mu \begin{bmatrix} 0 \\ u (\partial_x v + \partial_y u) + v [2\partial_y v + \lambda (\partial_x u + \partial_y v)] + (\gamma/\text{Pr}) \partial_y e \\ \partial_x v + \partial_y u \\ 2\partial_y v + \lambda (\partial_x u + \partial_y v) \end{bmatrix}, \quad (8)$$

where ∂_x and ∂_y refer to partial derivatives with respect to variables x and y , μ is the dynamic viscosity coefficient, Pr is the Prandtl number, and, using the Stokes hypothesis, $\lambda = -\frac{2}{3}$.

By multiplying the system of equations by an arbitrary smooth test function $\mathbf{v} = \{v_1, \dots, v_m\}$, and integrating by parts, we obtain the weak formulation

$$\int_{\Omega} \mathbf{v} \cdot \left(\mathbf{P}(\mathbf{w}) \frac{\partial \mathbf{w}}{\partial t} \right) dx - \int_{\Omega} \nabla \mathbf{v} : \mathbf{F}(\mathbf{w}, \nabla \mathbf{w}) dx + \int_{\partial \Omega} \mathbf{v} \otimes \mathbf{n} : \mathbf{F}(\mathbf{w}, \nabla \mathbf{w}) d\sigma = 0, \quad (9)$$

where \mathbf{F} is the sum of the convective and viscous flux functions and \mathbf{n} is the unit vector normal to the boundary. To discretize Eq. (9) we replace the solution \mathbf{w} and the test function \mathbf{v} with a finite element approximation \mathbf{w}_h and a discrete test function \mathbf{v}_h , respectively, where \mathbf{w}_h and \mathbf{v}_h belong to the space $\mathbf{V}_h^k = [\mathbb{P}_d^k(\mathcal{T}_h)]^m$. For each of the m equations of the system, and without loss of generality, we choose the set of test and shape functions in any element T coincident with the set $\{\phi\}$ of N_{dof}^T orthogonal and hierarchical basis functions in that element. With this choice each component $w_{h,j}$, $j = 1, \dots, m$, of $\mathbf{w}_h \in \mathbf{V}_h$ can be expressed, in terms of the elements of the global vector \mathbf{W} of unknown degrees of freedoms, as $w_{h,j} = \phi_l \mathbf{W}_{j,l}$, $l = 1, \dots, N_{dof}^T$, $\forall T \in \mathcal{T}_h$.

Finally, the DG discretization of the Navier–Stokes equations consists of seeking, for $j = 1, \dots, m$, the elements of \mathbf{W} such that

$$\begin{aligned} \sum_{T \in \mathcal{T}_h} \int_T \phi_i P_{j,k}(\mathbf{w}_h) \phi_l \frac{dW_{k,l}}{dt} d\mathbf{x} - \sum_{T \in \mathcal{T}_h} \int_T \frac{\partial \phi_i}{\partial x_n} F_{j,n}(\mathbf{w}_h, \nabla_h \mathbf{w}_h + \mathbf{r}(\llbracket \mathbf{w}_h \rrbracket)) d\mathbf{x} \\ + \sum_{\sigma \in \mathcal{F}_h} \int_{\sigma} \llbracket \phi_i \rrbracket_n \hat{F}_{j,n}(\mathbf{w}_h^{\pm}, (\nabla_h \mathbf{w}_h + \eta_F \mathbf{r}_F(\llbracket \mathbf{w}_h \rrbracket))^{\pm}) d\sigma = 0, \quad (10) \end{aligned}$$

for $i = 1, \dots, N_{\text{dof}}^T$. In the latter equation repeated indices imply summation over the ranges $k = 1, \dots, m, l = 1, \dots, N_{\text{dof}}^T, n = 1, \dots, d$. Here, the DG discretization of the viscous fluxes is based on the BR2 scheme, proposed in [15] and theoretically analyzed in [27] and [28]. According to this scheme, the viscous part of the numerical flux is given by

$$\hat{\mathbf{F}}_v(\mathbf{w}_h^{\pm}, (\nabla_h \mathbf{w}_h + \eta_F \mathbf{r}_F(\llbracket \mathbf{w}_h \rrbracket))^{\pm}) = \{\mathbf{F}_v(\mathbf{w}_h, \nabla_h \mathbf{w}_h + \eta_F \mathbf{r}_F(\llbracket \mathbf{w}_h \rrbracket))\}, \quad (11)$$

where the stability parameter η_F is defined according to [8]. In this work, the numerical approximation of the convective flux is given by the Roe’s scheme [29] with Harten’s entropy fix [30]. It is the upwind nature of this flux that guarantees the stability of the convective part of equations.

When dealing with DG formulations on agglomerated grids a technique for evaluating volume and surface integrals on arbitrarily shaped elements is required. A straightforward approach consists of computing the integrals on the agglomerated elements $T \in \mathcal{T}_h$ as the sum of the integrals on the composing sub-elements $E \in \mathcal{R}_T$. Integration on the sub-elements $E \in \mathcal{R}_T$ is then performed by means of standard quadrature rules defined on reference space elements.

This approach is very expensive when high-order approximations are considered, as the number of quadrature points becomes unacceptably large. To overcome this issue more efficient approaches to define quadrature rules on agglomerated elements have been proposed, for example in [31, 8, 3]. In these works, for a model equation, a detailed study of the gains in terms of quadrature points number and, for some cases, CPU time is reported.

The focus of this paper is the proof-of-concept of a non-standard adaptation procedure applied to fluid dynamics problems. Thus, in order to avoid any possible influence of numerical integration on the results, the quadrature points reduction strategies have not been used. The use and further development of efficient numerical integration rules for arbitrarily shaped elements as well as the assessment of their impact over the whole algorithm efficiency has been left to future work.

2.4. Time integration

By assembling together all the elemental contributions the system of ordinary differential equations governing the evolution in time of the discrete solution can be written as

$$\mathbf{M}_P(\mathbf{W}) \frac{d\mathbf{W}}{dt} + \mathbf{R}(\mathbf{W}) = \mathbf{0}, \quad (12)$$

where \mathbf{W} is the global vector of unknown degrees of freedom, \mathbf{M}_P is a global block diagonal matrix and $\mathbf{R}(\mathbf{W})$ is the vector of “residuals”. Eq. (12) can be integrated in time to the steady state by means of the Linearized Backward Euler (LBE) scheme. For steady solutions the choice of the time step can significantly affect the efficiency and the robustness of the method. To this end pseudo-transient continuation strategies, as those proposed in [32, 33], can be considered to enhance the solver performance. As recently shown by Leicht and Wallraff *et al.* in [34, 35] the steady state convergence can be greatly accelerated by using multigrid techniques. In this paper we apply h -multigrid to the non-linear system directly by means of the so-called Full Approximation Scheme (FAS). The non-linear problems of the fine-to-coarse levels are discretized by means of a LBE scheme where linear systems are solved with a Generalized Minimal RESidual (GMRES) method preconditioned with linear h -multigrid. An iterative line-Jacobi scheme is employed as linear smoother and coarse level solver within the multigrid [34, 35].

3. ADJOINT-BASED ERROR ESTIMATION

The adjoint problem provides a technique for estimating the error on a functional of interest. In this paper this estimation is used to drive an adaptive mesh agglomeration strategy to reduce the contribution of the spatial discretization error on the functional, see [14, 5].

3.1. Formulation of the adjoint problem

In this section we briefly introduce the use of the adjoint problem for the definition of a target-oriented local error estimator. Let $J(\mathbf{W})$ denote an objective functional of interest, *e.g.* the lift coefficient, computed from the discrete vector \mathbf{W} of the solution's degrees of freedom. For clarity we use the formulation as discrete sensitivity for the algebraic system of discretized equations rather than a variational formulation. Both approaches yield the same discrete adjoint problem due to the adjoint consistency of the discretization (including boundary conditions).

In the following the vector of degrees of freedom related to a k -th degree polynomial approximation of the solution $\mathbf{w}_k \in \mathbf{V}^k$ is denoted as \mathbf{W}_k . $J_{k+1}(\bullet)$ and $\mathbf{R}_{k+1}(\bullet)$ are the functional and the vector of the residuals related to a $\mathbb{P}_d^{k+1}(\mathcal{T}_h)$ spatial discretization computed with the \bullet vector.

The objective functional, formally obtained by increasing the degree of the discretization from k to $k+1$, is approximated to avoid the computational expense of solving a higher degree DG discretization of the non-linear system of equations, see also [5, 13]. The approximation relies on the Taylor series expansion of $J_{k+1}(\mathbf{W}_{k+1})$ about the L_2 -projection \mathbf{w}_k^{k+1} of the solution \mathbf{w}_k on \mathbf{V}^{k+1}

$$J_{k+1}(\mathbf{W}_{k+1}) \approx J_{k+1}(\mathbf{W}_k^{k+1}) + \left(\frac{\partial J_{k+1}}{\partial \mathbf{W}_{k+1}} \right)_{\mathbf{W}_k^{k+1}} (\mathbf{W}_{k+1} - \mathbf{W}_k^{k+1}), \quad (13)$$

similarly the vector of the residuals is expanded about \mathbf{W}_k^{k+1}

$$\mathbf{R}_{k+1}(\mathbf{W}_{k+1}) \approx \mathbf{R}_{k+1}(\mathbf{W}_k^{k+1}) + \left[\frac{\partial \mathbf{R}_{k+1}}{\partial \mathbf{W}_{k+1}} \right]_{\mathbf{W}_k^{k+1}} (\mathbf{W}_{k+1} - \mathbf{W}_k^{k+1}). \quad (14)$$

In practice, when orthogonal and hierarchical modal bases are used, the degrees of freedom of the projected solution are the same of the low-order one with null higher-order modes [36].

Since the unknown \mathbf{W}_{k+1} is the solution of the higher-order discretization of the non-linear problem we have $\mathbf{R}_{k+1}(\mathbf{W}_{k+1}) = \mathbf{0}$. Thus, by re-arranging Eq. (14) we obtain the following error representation

$$\mathbf{W}_{k+1} - \mathbf{W}_k^{k+1} \approx - \left[\frac{\partial \mathbf{R}_{k+1}}{\partial \mathbf{W}_{k+1}} \right]_{\mathbf{W}_k^{k+1}}^{-1} \mathbf{R}_{k+1}(\mathbf{W}_k^{k+1}). \quad (15)$$

The approximation of $J_{k+1}(\mathbf{W}_{k+1})$ can be obtained by substituting Eq. (15) into Eq. (13)

$$J_{k+1}(\mathbf{W}_{k+1}) \approx J_{k+1}(\mathbf{W}_k^{k+1}) - \left(\frac{\partial J_{k+1}}{\partial \mathbf{W}_{k+1}} \right)_{\mathbf{W}_k^{k+1}} \left[\frac{\partial \mathbf{R}_{k+1}}{\partial \mathbf{W}_{k+1}} \right]_{\mathbf{W}_k^{k+1}}^{-1} \mathbf{R}_{k+1}(\mathbf{W}_k^{k+1}), \quad (16)$$

where $\mathbf{R}_{k+1}(\mathbf{W}_k^{k+1}) \neq \mathbf{0}$ because the projected solution does not solve the equations system discretized on the richer polynomial space. This eliminates the unknown higher-order non-linear solution \mathbf{W}_{k+1} .

In order to avoid the explicit inversion of the Jacobian, the adjoint variable λ is then defined as the solution of the following discrete problem

$$\left[\frac{\partial \mathbf{R}_{k+1}}{\partial \mathbf{W}_{k+1}} \right]_{\mathbf{W}_k^{k+1}}^t (\lambda_{k+1})_{\mathbf{W}_k^{k+1}} = \left(\frac{\partial J_{k+1}}{\partial \mathbf{W}_{k+1}} \right)_{\mathbf{W}_k^{k+1}}^t, \quad (17)$$

where the superscript t indicates the transpose operator. By substituting Eq. (17) into Eq. (16) and re-arranging,

$$J_{k+1}(\mathbf{W}_{k+1}) - J_{k+1}(\mathbf{W}_k^{k+1}) \approx - (\lambda_{k+1})_{\mathbf{W}_k^{k+1}}^t \mathbf{R}_{k+1}(\mathbf{W}_k^{k+1}), \quad (18)$$

we have an estimate of the error of approximating the functional $J_{k+1}(\mathbf{W}_{k+1})$ with $J_{k+1}(\mathbf{W}_k^{k+1})$ as

$$e = -(\lambda_{k+1})_{\mathbf{W}_k^{k+1}}^t \mathbf{R}_{k+1}(\mathbf{W}_k^{k+1}). \quad (19)$$

Since basis functions are locally defined on each element, e can be decomposed as

$$e = \sum_{T \in \mathcal{T}_h} e^T \quad \text{with} \quad e^T = -(\lambda_{k+1})_{\mathbf{W}_k^{k+1}}^{Tt} \mathbf{R}_{k+1}^T(\mathbf{W}_k^{k+1}), \quad (20)$$

where the superscript T indicates that quantities are local to the mesh element. The local error indicator e^T is represented by the inner product of the local primal residuals with the discrete adjoint solution and can be used to drive the agglomeration-based mesh adaptation algorithm towards an accurate and efficient approximation of $J(\mathbf{W})$.

In this work for the solution of the linear system of Eq. (17) we rely on the same solver as employed for the linearized systems arising in the non-linear solver, *i.e.*, the system is solved by means of a GMRES Krylov subspace method which is preconditioned by a linear h -multigrid method, which in turn uses an iterative line-Jacobi approach as smoother and coarse level solver [34, 35]. In the numerical examples, the adjoint problem is considered converged when the linear residual norm has been reduced by eight orders of magnitude.

Enhancing the functional of interest with the adjoint problem As a first test to assess the effectiveness of the adjoint problem we consider the solution of the inviscid flow around the NACA0012 airfoil with $M_\infty = 0.5$ and $\alpha = 1^\circ$ on a very coarse grid made of 512 quadrangular elements with curved edges (quadratic). The lift coefficient C_L is chosen as the function of interest and we compare C_L^{k+1} , that is the lift coefficient computed with a $k+1$ degree polynomial approximation, with the lift coefficient extrapolated by means of the adjoint problem based on a k degree polynomial approximation. Since the error e evaluated by the adjoint problem approximates the difference between C_L^{k+1} and C_L^k , we should get

$$C_L^k + e \approx C_L^{k+1}. \quad (21)$$

The values of C_L^k and e are tabulated in Table I together with the effectiveness of the error estimation $\eta = e/(C_L^{k+1} - C_L^k)$. Being close to unity the η values indicate that the estimated errors are very close to the true errors, thus confirming that the extrapolation based on the adjoint problem can significantly improve the computed force coefficients.

3.2. The multi-target adjoint problem

Practical applications often require to accurately approximate multiple target quantities, consider, for example, the lift and drag aerodynamic coefficients, C_L and C_D . In this section, similar to Hartmann *et al.* [5], an adjoint problem targeting multiple force coefficients is presented. The idea is to compute an error distribution e_M^T that takes into account both of the C_L and the C_D sensitivities. The adjoint problem formulated for the C_L as functional of interest can be written as

$$\left[\frac{\partial \mathbf{R}_{k+1}}{\partial \mathbf{W}_{k+1}} \right]_{\mathbf{W}_k^{k+1}}^t (\lambda_{k+1}^{C_L})_{\mathbf{W}_k^{k+1}} = \left(\frac{\partial C_L^{k+1}}{\partial \mathbf{W}_{k+1}} \right)_{\mathbf{W}_k^{k+1}}^t, \quad (22)$$

while considering the drag coefficient, C_D , the problem reads

$$\left[\frac{\partial \mathbf{R}_{k+1}}{\partial \mathbf{W}_{k+1}} \right]_{\mathbf{W}_k^{k+1}}^t (\lambda_{k+1}^{C_D})_{\mathbf{W}_k^{k+1}} = \left(\frac{\partial C_D^{k+1}}{\partial \mathbf{W}_{k+1}} \right)_{\mathbf{W}_k^{k+1}}^t. \quad (23)$$

For the sake of notation compactness, we indicated as C_L^{k+1} and C_D^{k+1} the aerodynamic coefficients related to a $\mathbb{P}_d^{k+1}(\mathcal{T}_h)$ spatial discretization. To compute the sensitivity from C_L and C_D together,

the following linear combination is considered

$$(\lambda_{k+1}^M)_{\mathbf{W}_k^{k+1}} = \alpha (\lambda_{k+1}^{C_L})_{\mathbf{W}_k^{k+1}} + \beta (\lambda_{k+1}^{C_D})_{\mathbf{W}_k^{k+1}}. \quad (24)$$

The multi-target sensitivity λ_{k+1}^M could be trivially computed by solving the two adjoint problems of Eqs. (22) and (23). However, due to their linear nature, we can substitute Eqs. (22) and (23) into Eq. (24) and re-arrange the equation, thus obtaining a single linear problem to solve

$$\left[\frac{\partial \mathbf{R}_{k+1}}{\partial \mathbf{W}_{k+1}} \right]_{\mathbf{W}_k^{k+1}}^t (\lambda_{k+1}^M)_{\mathbf{W}_k^{k+1}} = \alpha \left(\frac{\partial C_L^{k+1}}{\partial \mathbf{W}_{k+1}} \right)_{\mathbf{W}_k^{k+1}}^t + \beta \left(\frac{\partial C_D^{k+1}}{\partial \mathbf{W}_{k+1}} \right)_{\mathbf{W}_k^{k+1}}^t. \quad (25)$$

The parameters α and β should be chosen so to ensure that the sensitivities of C_L and C_D are of same order of magnitude; in this work we simply set $\alpha = 1/C_L$ and $\beta = 1/C_D$. Following the same steps of Sec. 3.1, the elemental error estimate reads

$$e_M^T = - (\lambda_{k+1}^M)_{\mathbf{W}_k^{k+1}}^{Tt} \mathbf{R}_{k+1}^T (\mathbf{W}_k^{k+1}). \quad (26)$$

The error field resulting from evaluating e_M^T in each mesh element can be then employed to drive an adaptive mesh refinement algorithm.

4. ADAPTIVE AGGLOMERATION STRATEGIES DRIVEN BY THE ADJOINT PROBLEM

The goal of the mesh agglomeration strategy is to optimize the mesh step size according to the error estimator provided by the adjoint problem in order improve the accuracy on the functional of interest.

A fine mesh \mathcal{R} , designed to provide a suitable approximation Ω_h of the computational domain Ω , is required first. In this work the mesh density is chosen to ensure a high-fidelity representation of the airfoil geometry at the discrete level, which is of primary importance to correctly estimate the aerodynamic coefficients, and to provide a sufficient amount of flexibility in defining agglomerated elements over the whole domain. Nevertheless, as already remarked in Sec. 2.3, the grid density should not be pushed towards the extremely fine limit to avoid an uncontrolled growth of the numerical integration costs. The first agglomerated elements mesh is guessed by setting a uniform agglomeration factor over the fine grid, that is imposing $a^E = \mathcal{I}$ in each mesh element E . The iterative adaptation strategy is initialised with such grid, and, at each iteration loop $l = 1, \dots, \mathcal{L}$, the agglomeration factor is tweaked according to the error metric. In particular, the number of elements to be refined is a user-defined fraction \mathcal{G} of the total number of mesh elements and refinement is triggered according to the error metric $|e^T|$. Mesh refinement is achieved by reducing the agglomeration factor a^E . The steepness of the agglomeration factor reduction is controlled by the refinement factor \mathcal{N} . Clearly the functional of interest, the initial agglomeration factor \mathcal{I} , the number of adaptation loops \mathcal{L} , the fraction of elements marked for refinement \mathcal{G} , the refinement factor \mathcal{N} as well as the initial mesh density are user-defined parameters that might influence the refinement outcome.

In the following sub-sections we investigate several choices for \mathcal{N} , starting from the trivial choice of picking a constant \mathcal{N} value.

4.1. The adaptive algorithm - \mathcal{N} fixed

In this section we consider the adaptive mesh refinement strategy obtained choosing a constant \mathcal{N} , the same for every loop of the strategy. Since a^E is an upper bound for the number of sub-elements that can be clustered together with E to form an agglomerated element T , see Section 2.2, for some $E \in \mathcal{R}_T$ we might get $a^E > \text{card}(\mathcal{R}_T)$. As a consequence the agglomeration factor does not fully reflect the grid generated by the MGridGen library, see Figure 3. Hence we propose to update a^E with the actual number of element of the sub-elements contained in the agglomerated element T ,

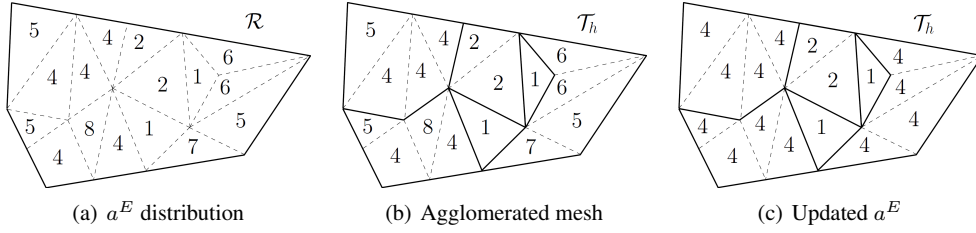


Figure 3. Effect of a^E updated with the actual number of agglomerated elements

i.e. $\text{card}(\mathcal{R}_T)$.

The adaptation strategy procedure is written below in pseudo-code

Algorithm 1 \mathcal{N} fixed adaptive algorithm, a^E updated

```

1:  $\{\mathcal{N}$  fixed adaptive algorithm,  $a^E$  updated $\}$ 
2: Fine mesh  $\mathcal{R}$ 
3:  $a^E = \mathcal{I}$ 
4: for  $l = 1$  to  $\mathcal{L}$  do
5:   mesh agglomeration
6:   for  $T \in \mathcal{T}_h$  do
7:     for  $E \in \mathcal{R}_T$  do
8:        $a^E = \text{card}(\mathcal{R}_T)$ 
9:     end for
10:  end for
11:  Solve direct problem
12:  Solve adjoint problem and compute the  $e^T$  distribution
13:  for  $T \in \mathcal{T}_h$  do
14:    if  $\text{POS}_T \geq (1 - \mathcal{G})\text{card}(\mathcal{T}_h)$  then
15:      for  $E \in \mathcal{R}_T$  do
16:         $a^E = \left\lceil \frac{a^E}{\mathcal{N}} \right\rceil$ 
17:      end for
18:    end if
19:  end for
20: end for

```

At each iteration l the direct problem is solved, the functional of interest is evaluated and the local error estimator e^T is computed $\forall T \in \mathcal{T}_h$ solving the adjoint problem (which requires polynomial enrichment). The elements are sorted in increasing order according to $|e^T|$ and numbered starting from zero to reflect the element position POS_T in the sorted elements vector. An element T is marked for refinement if $\text{POS}_T \geq (1 - \mathcal{G})\text{card}(\mathcal{T}_h)$. If refinement is triggered the agglomeration indicator is divided by \mathcal{N} (and approximated at the nearest upper integer, that is $a^E = \left\lceil \frac{a^E}{\mathcal{N}} \right\rceil$) thereby reducing the mesh step size in those regions whose lack of accuracy has the largest impact on the function of interest. The new agglomerated elements mesh is considered as the initial guess for the second loop, and so on until the desired numbers of loops is reached. The non-linear problems considered in this work are relatively easy to solve with the iterative solver employed. Consequently, the initial solution has a relatively small impact on the solution time. Thus, initialization for the non-linear problem is based on free-stream values in the whole domain rather than a projection/interpolation of the solution obtained in the last adaptation cycle. Nevertheless, such a projection can easily be obtained in the proposed framework due to the availability of a common fine mesh and should be implemented prior to any timing comparison.

4.1.1. Uniform versus adaptive mesh agglomeration In this section the convergence of the adaptive strategy towards a reference value of C_L is compared with the results obtained on a set of uniformly agglomerated meshes built on top of the fine mesh \mathcal{R} with $\mathcal{I} \in \{2, 4, 8, 16\}$, see Figure 4. All the computations are run with $k = 1$ and $k = 2$, corresponding to a second- and third-order discretizations respectively. As a reference value C_L^{*k} we consider the lift coefficient computed on the fine grid \mathcal{R} , composed of 26 406 triangular elements. This mesh has been generated with a 2D high-order version of a fully automated hybrid mesh generator based on the advancing-Delaunay strategy [37]. The reference C_L values are $C_L^{*1} = 0.1404786875$ and $C_L^{*2} = 0.1410151963$ for the \mathbb{P}^1 and \mathbb{P}^2 solutions, respectively.

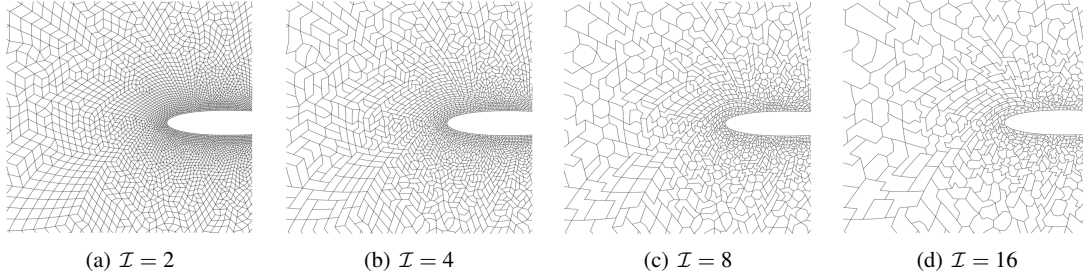


Figure 4. NACA0012: Uniformly agglomerated meshes built on top of the fine mesh \mathcal{R} of 26 406 elements

The error \mathcal{E} with respect to the reference solution is computed as follows

$$\mathcal{E} = \left| \frac{C_L^k - C_L^{*k}}{C_L^{*k}} \right|, \quad (27)$$

and plotted, for both \mathbb{P}^1 and \mathbb{P}^2 , against the corresponding grid cardinality in Figure 5. We remark that, these errors are measured w.r.t. the C_L value obtained with the same polynomial degree on the given underlying fine mesh. Although in one hand, this can be considered as misleading since not considering the “true” asymptotic mesh converged value, on the other hand this is the correct measure of what the agglomeration-based adaptive method is able to achieve (by design) in terms of accuracy. In this first test we set $\mathcal{G} = 10\%$.

In Figure 5, the red segments connect the C_L errors associated with two subsequent adaptively refined grids resulting from the agglomeration-based adaptive strategy. In contrast the green line displays the C_L errors associated with the uniformly agglomerated meshes. For both the polynomial approximations here considered the adaptation strategy manages to reach a smaller error than uniformly agglomerated meshes with the same number of elements, demonstrating its effectiveness. For both the \mathbb{P}^1 and \mathbb{P}^2 computations the adaptive mesh tops at 4 000 elements after 15 adaptive loops, that is far less than the $\approx 26\,000$ elements of the fine underlying mesh \mathcal{R} . Comparing the adaptively refined mesh (resulting from 15 refinement iterations) with the uniformly agglomerated mesh obtained setting $\mathcal{I} = 2$, made of $\approx 14\,000$ elements, the error on the lift coefficient is one order of magnitude less ($\approx 10^{-3}$ instead of $\approx 10^{-2}$) for \mathbb{P}^1 and two orders of magnitude less ($\approx 10^{-5}$ instead of $\approx 10^{-3}$) for \mathbb{P}^2 .

4.1.2. Influence of the refinement fraction \mathcal{G} on the adaptive strategy In this section we assess the influence of refinement fraction \mathcal{G} on the performance of the adaptation strategy. The number of loops is kept fixed, setting $\mathcal{L} = 15$, while the refinement percentage is varied: $\mathcal{G} \in \{5\%, 10\%, 15\%, 20\%\}$. As displayed in Figure 6 all the simulations reach approximately the same error level of $\approx 10^{-6}$ but, as expected, the final number of elements increases while increasing \mathcal{G} . Since the convergence towards the final error plateau is not significantly improved increasing \mathcal{G} the numerical results suggest that $\mathcal{G}=5\%$ and $\mathcal{G}=10\%$ are the most valuable choices. Nevertheless, the error history observed at $\mathcal{G}=5\%$ suggests that more refinement iterations would be required to converge towards a stable error value, discouraging a further reduction of the refinement fraction.

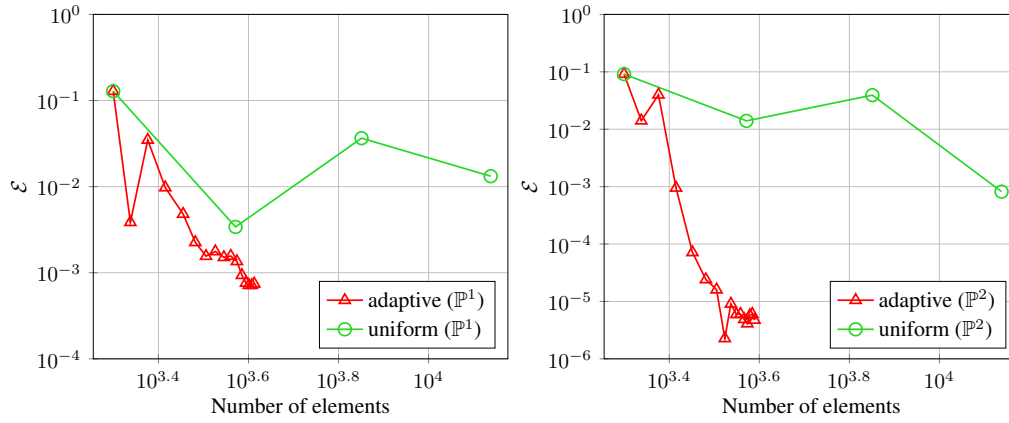


Figure 5. NACA0012 - $M_\infty = 0.5$, $\alpha = 1^\circ$, $\mathbb{P}^{1,2}$ solutions: Lift coefficient error vs. mesh cardinality, \mathcal{R} is made of 26 406 triangular elements

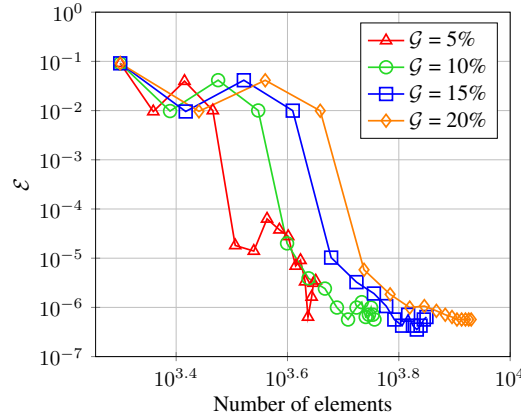


Figure 6. NACA0012 - $M_\infty = 0.5$, $\alpha = 1^\circ$, \mathbb{P}^2 solution: Influence of \mathcal{G} on adaptation performance, \mathcal{R} is made of 26 406 triangular elements

4.2. The adaptive algorithm - variable \mathcal{N}

Reducing the agglomeration factor by a fixed amount for a fixed fraction of the mesh elements has shown to be effective but more sophisticated strategies might be devised. With the aim of refining or coarsening every element depending on its adjoint based error estimate, we seek a relationship between element refinement or coarsening and the expected error reduction. Dividing the agglomeration factor by \mathcal{N} should cause a reduction of the element size proportional to $\frac{1}{\mathcal{N}}^{\frac{1}{d}}$. Considering the convergence properties of the DG discretization, the target error ϵ for the refined agglomerated element can thus be expressed as a function of the actual error e^T as follows

$$\epsilon = \frac{e^T}{\mathcal{N}^{\frac{k+1}{d}}}, \quad (28)$$

where d is the spatial dimension, *i.e.*, $d = 2$ for the example considered here. According to Eq. (28), in each element T , \mathcal{N}^T can be computed as follows

$$\mathcal{N}^T = \left(\frac{|e^T|}{\epsilon} \right)^{\frac{d}{k+1}}. \quad (29)$$

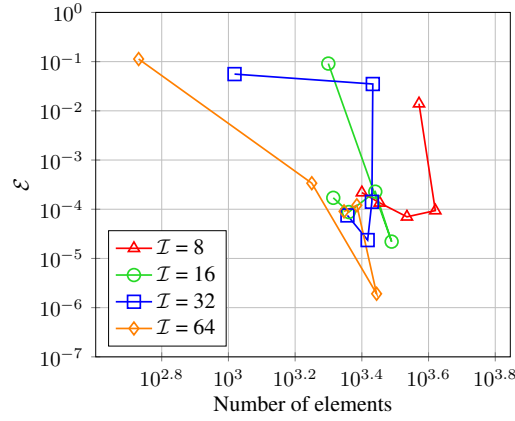


Figure 7. NACA0012 - $M_\infty = 0.5$, $\alpha = 1^\circ$, \mathbb{P}^2 solution: Influence of starting agglomeration factor \mathcal{I} on the variable \mathcal{N} strategy with coarsening enabled (Algorithm 2), \mathcal{R} is made of 26 406 triangular elements

We set the target error ϵ to a fraction γ of the average global error computed by the adjoint problem

$$\epsilon = \gamma \bar{e} = \gamma \frac{\sum_{T \in \mathcal{T}_h} e^T}{\text{card}(\mathcal{T}_h)}. \quad (30)$$

When $\epsilon > |e^T|$ Eq. (29) predicts $\mathcal{N}^T < 1$, thus promoting the generation via re-agglomeration of larger elements and thus enabling coarsening.

The pseudo-code of a simple possible adaptation algorithm is reported below

Algorithm 2 variable \mathcal{N} adaptive algorithm (coarsening enabled)

```

1: Generate the fine mesh  $\mathcal{R}$ 
2:  $a^E = \mathcal{I}$ 
3: for  $l = 1$  to  $\mathcal{L}$  do
4:   Agglomerate the mesh
5:   for  $T \in \mathcal{T}_h$  do
6:     for  $E \in \mathcal{R}_T$  do
7:        $a^E = \text{card}(\mathcal{R}_T)$ 
8:     end for
9:   end for
10:  Solve direct problem
11:  Solve adjoint problem and compute the error field  $e^T$ , for each  $T \in \mathcal{T}_h$ 
12:  for  $T \in \mathcal{T}_h$  do
13:     $\mathcal{N}^T = \left( \frac{|e^T|}{\epsilon} \right)^{\frac{d}{k+1}}$ 
14:    for  $E \in \mathcal{R}_T$  do
15:       $a^E = \min \left( \left\lceil \frac{a^E}{\mathcal{N}^T} \right\rceil, 2a^E \right)$ 
16:    end for
17:  end for
18: end for
```

We assess the performance of this adaptation algorithm by running four adaptation steps ($\mathcal{L} = 4$) with a fixed $\gamma=1/10$ value and varying $\mathcal{I} \in \{8, 16, 32, 64\}$. The results shown in Figure 7 are encouraging. The algorithm exhibits a behavior that is fairly independent of the initial distribution of the degrees of freedom. The adaptation strategy leads to similar results both in terms of error level and number of elements as tabulated in Table II.

However, after a drastic reduction of the error in the first or first two adaptation cycles, the number of elements decreases and the error correspondingly grows again. This can be attributed to the fact that the ability for refinement is limited by the existing underlying mesh resolution. Once this resolution is reached when only a single element is “agglomerated” these elements can continue to contain the highest (estimated) error and thus be further selected for refinement, but this will have no effect. In contrast to that, coarsening of lower error regions will continue, thus yielding a decreasing mesh size in combination with a slight rise of the resulting error level.

This effect is undesired, as we aim at achieving almost the full accuracy that can be obtained on the underlying grid on an agglomerated mesh with fewer elements, thus limiting the overall effort and size of the problem to be considered. In this process, reducing the accuracy via using smaller meshes later in the adaptive cycle is counter-productive. For this reason coarsening was disabled in this paper, meaning that if $\mathcal{N}^T < 1$, then a^E will not be modified. A thorough investigation on the potential of coarsening has been left to future work.

According to our numerical experiments the following heuristics were introduced to improve the performance of the algorithm:

- If $\mathcal{N}^T > 1$ and $POS_T \geq (1 - \mathcal{G})\text{card}(\mathcal{T}_h)$ then \mathcal{N}^T is forced to be at least 2.
- If $\mathcal{N}^T > 1$, $POS_T < (1 - \mathcal{G})\text{card}(\mathcal{T}_h)$ and $a^E > 1$ then a^E is reduced by at least 1.

All the computations presented in the remainder of the paper were obtained with this last adaptation strategy.

The pseudo-code is reported below

Algorithm 3 variable \mathcal{N} adaptive algorithm (refinement only)

```

1: Generate the fine mesh  $\mathcal{R}$ 
2:  $a^E = \mathcal{I}$ 
3: for  $l = 1$  to  $\mathcal{L}$  do
4:   Agglomerate the mesh
5:   for  $T \in \mathcal{T}_h$  do
6:     for  $E \in \mathcal{R}_T$  do
7:        $a^E = \text{card}(\mathcal{R}_T)$ 
8:     end for
9:   end for
10:  Solve direct problem
11:  Solve adjoint problem and compute the error field  $e^T$ , for each  $T \in \mathcal{T}_h$ 
12:  for  $T \in \mathcal{T}_h$  do
13:     $\mathcal{N}^T = \left( \frac{|e^T|}{\epsilon} \right)^{\frac{d}{k+1}}$ 
14:    if  $\mathcal{N}^T > 1$  then
15:      for  $E \in \mathcal{R}_T$  do
16:        if  $POS_T \geq (1 - \mathcal{G})\text{card}(\mathcal{T}_h)$  then
17:           $a^E = \min \left( \left\lceil \frac{a^E}{\mathcal{N}^T} \right\rceil, \left\lceil \frac{a^E}{2} \right\rceil \right)$ 
18:        else if  $a^E > 1$  then
19:           $a^E = \min \left( \left\lceil \frac{a^E}{\mathcal{N}^T} \right\rceil, a^E - 1 \right)$ 
20:        end if
21:      end for
22:    else
23:       $a^E = a^E$ 
24:    end if
25:  end for
26: end for
```

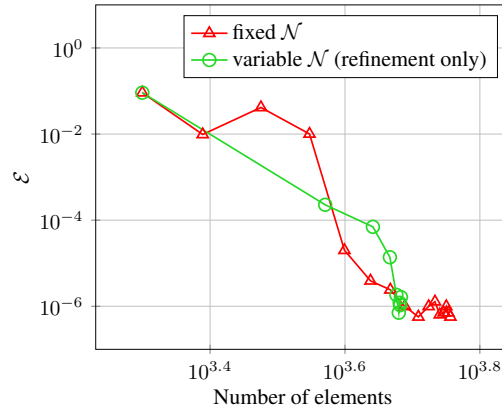


Figure 8. NACA0012 - $M_\infty = 0.5$, $\alpha = 1^\circ$, \mathbb{P}^2 solution: Fixed \mathcal{N} vs. variable \mathcal{N} strategy with coarsening disabled, 26 406 triangular elements mesh

The results reported in Figure 8 testify that this variable \mathcal{N} strategy with coarsening disabled leads to significant improvement with respect to the fixed \mathcal{N} strategy (Algorithm 1). The C_L error reaches 10^{-6} , the same order of magnitude of the fixed \mathcal{N} strategy, but with a smaller number of refinement steps ($\mathcal{L} = 8$) and the resulting adaptively refined grid has less agglomerated elements.

The distributions of the error indicator e^T and of the resulting agglomeration factor a^E together with the resulting adaptively agglomerated grid \mathcal{T}_h , for the first as well as the last refinement iteration, are shown in Figure 9. Comparing the initial and final error fields, it is evident that the adaptive algorithm allows to obtain a much more uniform distribution of the error over the computational domain. Nevertheless, high errors are still present in a cluster of elements located in the immediate proximity of the nose of the airfoil. Since the agglomeration factor has reached its lowest limit, that is $a^E = 1$, no further refinement is possible for these elements. This occurrence can be considered an inherent limitation of the agglomeration-based adaptive mesh refinement strategy, but it also provides an inherent way to avoid an excessive growth of the number of elements. To further enhance the accuracy a finer mesh \mathcal{R} could be considered as starting point of the algorithm.

4.2.1. Influence of the starting agglomeration factor \mathcal{I} In the following, initialization by means of a uniform agglomeration factor \mathcal{I} is investigated to evaluate the sensitivity to such parameter. Whatever $\mathcal{I} \in \{8, 16, 32, 64\}$ is chosen, after 8 adaptation loops the numerical solutions reach a similar accuracy with an error level in the range $10^{-6} - 10^{-7}$, see Figure 10. Moreover, the cardinality of the adaptively-agglomerated meshes is similar for $\mathcal{I} \in \{16, 32, 64\}$ and slightly higher for 8 due to the choice of disabling coarsening. This means that for a given high enough \mathcal{I} value the resulting efficiency of the algorithm is largely independent of the actual \mathcal{I} . Compared to the case considered previously, cf. Figure 7, the detrimental effect of coarsening is avoided, as intended. Instead, the algorithm approaches a certain “optimal” mesh quickly and then remains at an almost constant level of mesh elements and resulting accuracy.

4.2.2. Uniform versus adaptive mesh agglomeration In this sub-section the adaptation algorithm is applied to a finer mesh consisting of 114 532 triangular elements with quadratic edges. The convergence of the adaptive strategy towards the reference value of C_L ($C_L^{*2} = 0.1407924964$) is compared with the results obtained on a set of uniformly agglomerated meshes obtained on the top of the fine mesh \mathcal{R} with $\mathcal{I} \in \{2, 4, 8, 16\}$. All the computations are run with a \mathbb{P}^2 approximation of the solution. As shown in Figure 11, after only two loops an error of $\approx 10^{-6}$ is attained. Further cycles of the adaptation algorithm show very little variation in the resulting mesh and accuracy. Uniform agglomeration fails to reach this error level, even using a higher number of elements. Comparing with $\mathcal{I} = 2$ the error is three orders of magnitude lower with approximately 1/3 of the degrees of

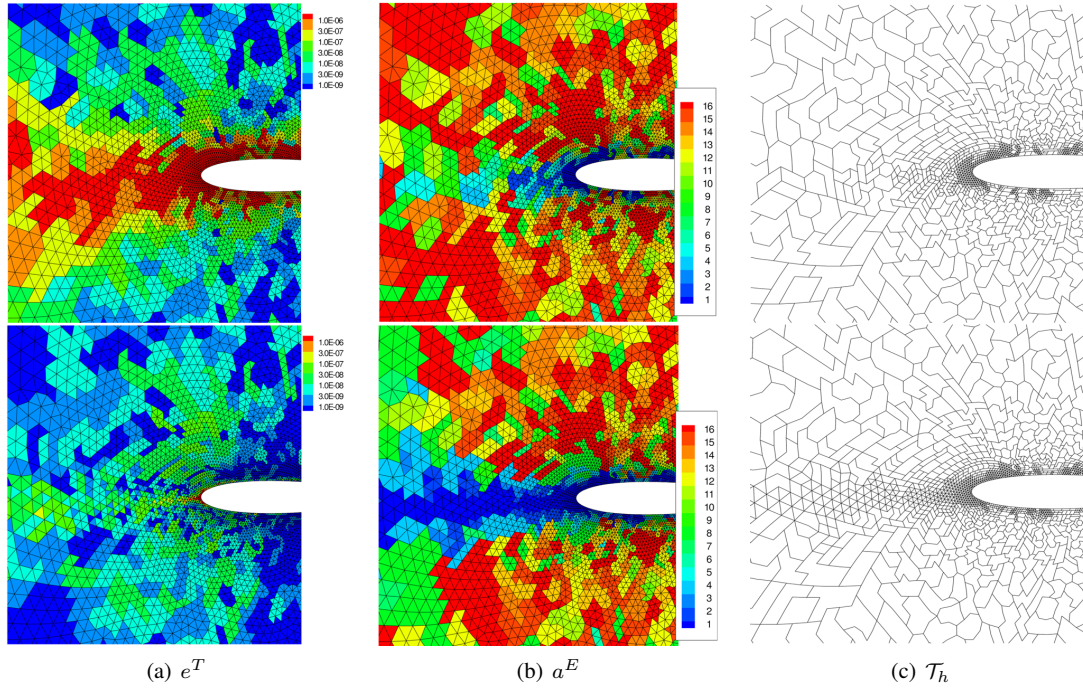


Figure 9. NACA0012 - $M_\infty = 0.5$, $\alpha = 1^\circ$, \mathbb{P}^2 solution: Distributions of the error indicator e^T and of the resulting agglomeration factor a^E together with the resulting adaptively agglomerated grid \mathcal{T}_h , for the first (top) and the last (bottom) refinement iteration of the \mathcal{N} -variable strategy with $\gamma=1/10$. (\mathcal{R} is a 26 406 triangular elements mesh)

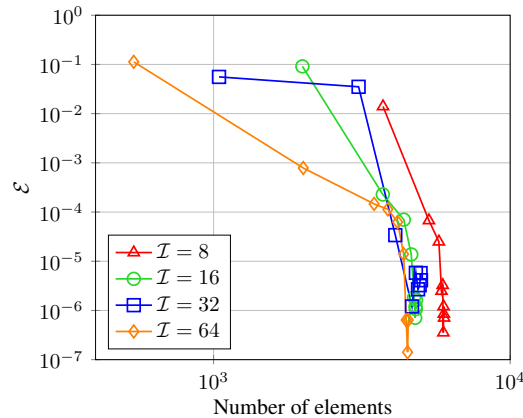


Figure 10. NACA0012 - $M_\infty = 0.5$, $\alpha = 1^\circ$, \mathbb{P}^2 solution: Influence of starting agglomeration factor \mathcal{I} on the variable \mathcal{N} strategy, \mathcal{R} is made of 26 406 triangular elements

freedom. The error associated to adaptive mesh refinement strategies, after a steep decrease in the first couple of iterations, stagnates around a fixed error level and the number of elements does not increases significantly. In practice, by comparing to subsequent iterations, it would be easy to detect convergence of the algorithm and arrest the adaptation algorithm when no further improvements are achievable.

The distributions of the error indicator e^T and of the resulting agglomeration factor a^E together with the resulting adaptively agglomerated grid \mathcal{T}_h , for the first as well as the last refinement iteration are shown in Figure 12. Once again the adaptive algorithm allows to obtain a much more uniform distribution of the error over the computational domain. Moreover, thanks to the use of a finer

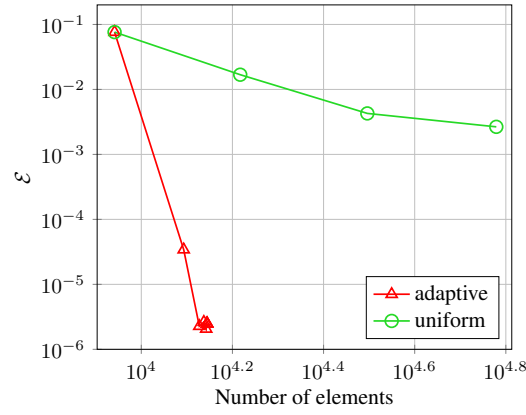


Figure 11. NACA0012 - $M_\infty = 0.5$, $\alpha = 1^\circ$, \mathbb{P}^2 solution: Error on the lift coefficient C_L over adaptively agglomerated meshes and uniformly agglomerated meshes, \mathcal{R} is made of 114 532 triangular elements

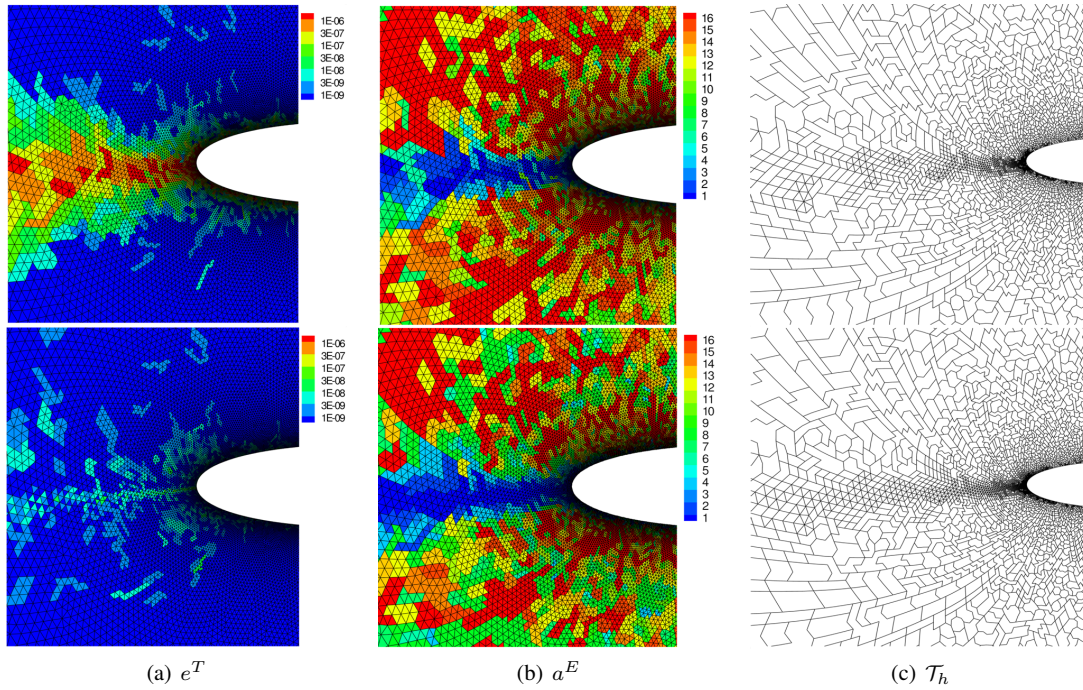


Figure 12. NACA0012 - $M_\infty = 0.5$, $\alpha = 1^\circ$, \mathbb{P}^2 solution: Distributions of the error indicator e^T and of the resulting agglomeration factor a^E together with the resulting adaptively agglomerated grid \mathcal{T}_h , for the first (top) and the last (bottom) refinement iteration of the \mathcal{N} -variable strategy with $\gamma=1/10$. (\mathcal{R} is a 114 532 triangular elements mesh)

grid, the high error regions located at the nose of the airfoil, see Figure 9, have disappeared almost completely.

5. MULTI-TARGET ADJOINT PROBLEM APPLICATION

In this chapter we apply the variable \mathcal{N} adaptation strategy to viscous multi-target computations. In this configuration, since we also take into account the viscous terms in the Navier–Stokes equations, it makes sense to optimize the mesh with respect to the computation of both the lift and drag

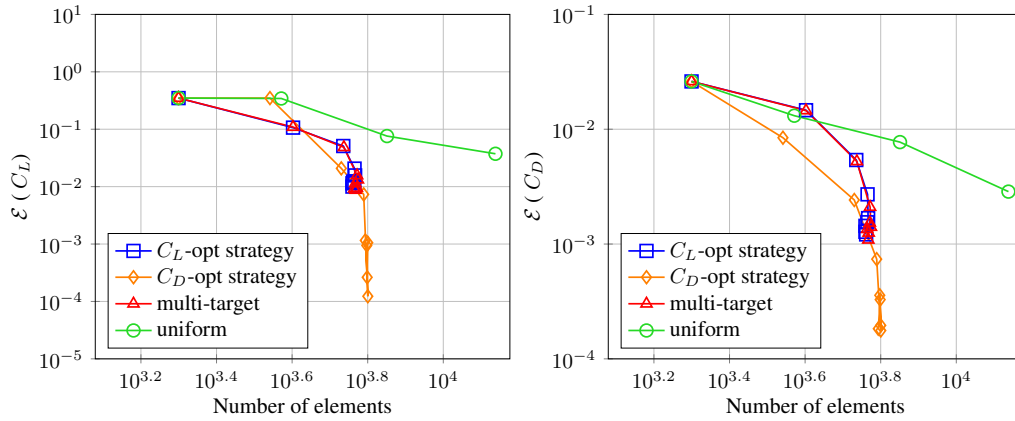


Figure 13. NACA0012 - $M_\infty = 0.5$, $\alpha = 1^\circ$, $Re_c = 5000$, \mathbb{P}^2 solution: C_L and C_D errors with respect to reference values for different single- and multi-target adjoint strategies, \mathcal{R} is made of 26 406 triangular elements

coefficients, C_L and C_D , simultaneously. In addition to the multi-target adjoint problem for C_L and C_D we also consider adjoint problems for C_L and C_D individually and compare the accuracy of these three approaches.

We consider the flow around the NACA0012 airfoil with $M_\infty = 0.5$, $\alpha = 1^\circ$ and chord-based $Re_c = 5000$. The fine computational grids are the same as those considered for inviscid test cases in Secs. 4.1 and 4.2, that is triangular elements grids with quadratic edges consisting of 26 406 and 114 532 elements. For all the computations we rely on a third-order \mathbb{P}^2 DG discretization. As for the adaptation strategy we perform $\mathcal{L} = 8$ adaptive mesh optimization iterations, starting from a uniformly agglomerated grid computed with the initial agglomeration factor set to $\mathcal{I} = 16$. C_L and C_D values computed on adaptively refined meshes are compared with reference values computed on the fine underlying grids \mathcal{R} , see Table III.

The multi-target optimization strategy performs well when the finer grid is considered, see Figure 14, while it basically replicates the results obtained with C_L as the sole target on the 26 406 elements grid, as shown Figure 13. This relatively similar behavior might be due to the fact that C_L and C_D are similar enough, both are normalized forces acting on the airfoil, and as such both require reasonable resolution of the boundary layer and stagnation point as well as the stagnation stream line region.

Compared with the inviscid simulations reported in Sec. 4 the adapted meshes are more refined in the boundary layer and in the wake region as shown in Figure 15.

6. CONCLUSIONS

In this work a non-standard approach to the mesh adaptation has been considered. The approach is based on the agglomeration, driven by an adjoint-based error estimator, of the elements of a fine enough underlying mesh made of standard elements. Different adaptation algorithms have been presented and their performance assessed on both inviscid and viscous test cases. These algorithms, independent from the polynomial degree of the space approximation, revealed an appealing coupling between the agglomeration-based adaptation strategy and the high-order DG framework.

While other adaptive strategies typically aim at reducing the error of a discrete approximation w.r.t. a (mesh-)asymptotic value, the proposed method is limited to the accuracy obtainable with the given discretization on the underlying fine mesh. The algorithm aims in fact at reaching, on an agglomerated coarser mesh, an error value very close to that obtainable accuracy. The numerical examples demonstrate that this aim can be achieved with a considerable reduction in the resulting problem size. After very few adaptive iterations the discrete error approaches the resolution limit

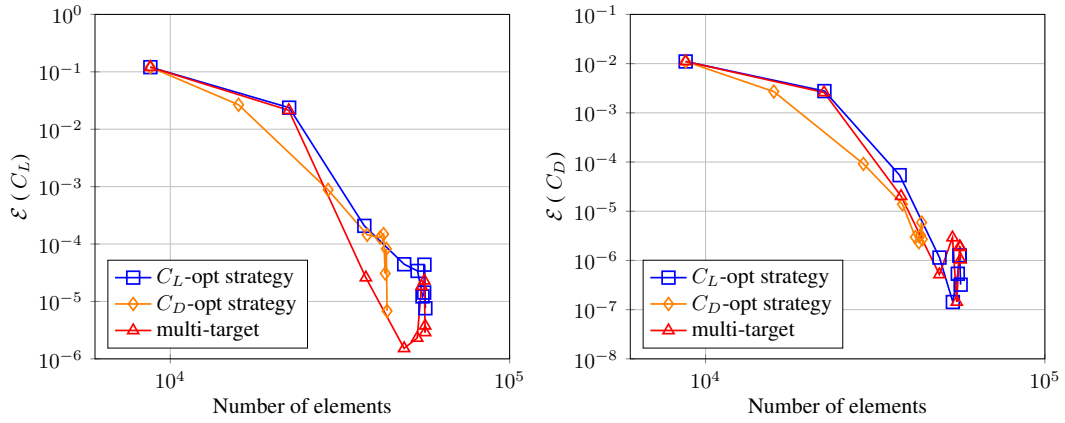


Figure 14. NACA0012 - $M_\infty = 0.5$, $\alpha = 1^\circ$, $Re_c = 5\,000$, \mathbb{P}^2 solution: C_L and C_D errors with respect to reference values for different single- and multi-target adjoint strategies, \mathcal{R} is made of 114 532 triangular elements

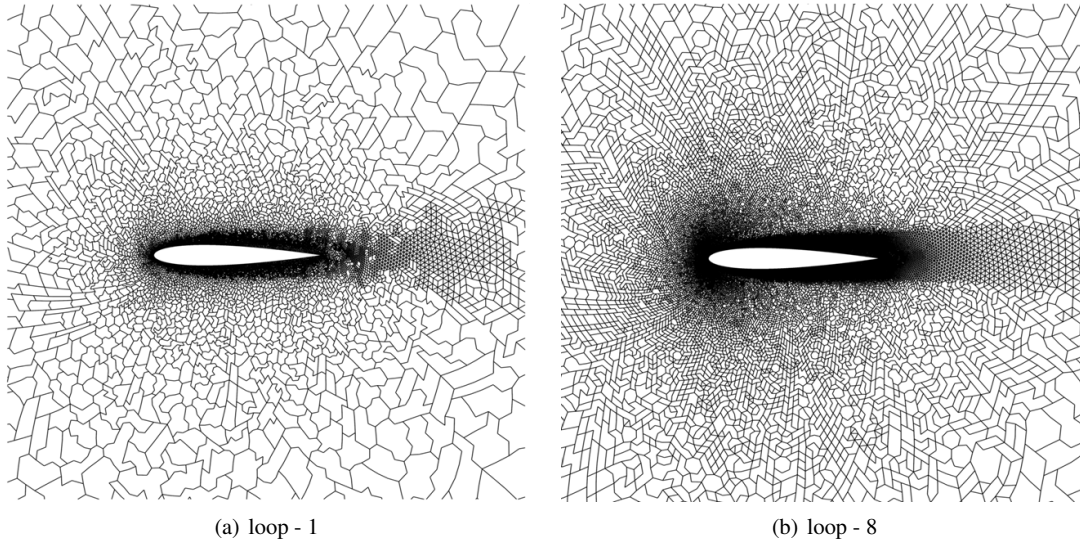


Figure 15. NACA0012 - $M_\infty = 0.5$, $\alpha = 1^\circ$, $Re_c = 5\,000$: Details of the adaptively agglomerated meshes resulting from a multi-target strategy, first (left) and last (right) iteration of the adaptation strategy

of the fine mesh, *i.e.*, the error of the coarse mesh solution w.r.t. the fine mesh solution is orders of magnitude smaller than the remaining error of the fine mesh solution w.r.t. the asymptotic value.

In this work, problems have been considered for which isotropic elements and agglomerates are appropriate. Other flow problems typically require highly stretched anisotropic elements, for example when considering boundary layers in high Reynolds number viscous flows. For these problems, the present strategy of having a uniform agglomeration factor that does not contain directional informations might be inappropriate, in particular as the MGridGen library aims at producing agglomerates of isotropic shapes. Corresponding extensions of the proposed methodology are an interesting topic of future work.

REFERENCES

1. G. J. Gassner, F. Lörcher, C.-D. Munz, and J. S. Hesthaven. Polymorphic nodal elements and their application in discontinuous Galerkin methods. *J. Comput. Phys.*, 228:1573–1590, 2009.

2. F. Bassi, L. Botti, A. Colombo, and S. Rebay. Agglomeration based discontinuous Galerkin discretization of the Euler and Navier–Stokes equations. *Comput. Fluids*, 61(0):77 – 85, 2012.
3. F. Bassi, L. Botti, and A. Colombo. Agglomeration-based physical frame dG discretizations: An attempt to be mesh free. *Mathematical Models and Methods in Applied Sciences*, 24(8):1495–1539, 2014.
4. K.J. Fidkowski and D.L. Darmofal. Review of output-based error estimation and mesh adaptation in computational fluid dynamics. *AIAA Journal*, 49(4):673–694, 2011.
5. R. Hartmann, J. Held, and T. Leicht. Adjoint-based error estimation and adaptive mesh refinement for the RANS and k - ω turbulence model equations. *Journal of Computational Physics*, (230):4268–4284, 2011.
6. T. Leicht and R. Hartmann. Error estimation and anisotropic mesh refinement for 3d laminar aerodynamic flow simulations. *Journal of Computational Physics*, (229):7344–7360, 2010.
7. V. Dolejší and J. Felcman. Anisotropic mesh adaptation for numerical solution of boundary value problems. *Numerical Methods for Partial Differential Equations*, 20:576–608, 2004.
8. F. Bassi, L. Botti, A. Colombo, D. A. Di Pietro, and P. Tesini. On the flexibility of agglomeration based physical space discontinuous Galerkin discretizations. *Journal of Computational Physics*, pages 45–65, 2012.
9. D. A. Di Pietro and R. Specogna. An a posteriori-driven adaptive mixed high-order method with application to electrostatics. *Journal of Computational Physics*, 326:35 – 55, 2016.
10. J. Collis and P. Houston. Adaptive Discontinuous Galerkin Methods on Polytopic Meshes. In Giulio Ventura and Elena Benvenuti, editors, *Advances in Discretization Methods: Discontinuities, Virtual Elements, Fictitious Domain Methods*, pages 187–206. Springer International Publishing, Cham, 2016.
11. F. Bassi and S. Rebay. High-order accurate discontinuous finite element solution of the 2D Euler equations. *J. Comput. Phys.*, 138:251–285, 1997.
12. R. Hartmann. Adaptive discontinuous Galerkin methods with shock-capturing for the compressible Navier-Stokes equations. *Int. J. Numer. Meth. Fluids*, 51:1131–1156, 2006.
13. R. Hartmann and P. Houston. Adaptive discontinuous Galerkin finite element methods for the compressible Euler equations. *J. Comput. Phys.*, 183:508–532, 2002.
14. Li Wang and Dimitri J. Mavriplis. Adjoint-based h - p adaptive discontinuous Galerkin methods for the 2D compressible Euler equations. *J. Comput. Phys.*, 228:7643–7661, November 2009.
15. F. Bassi, S. Rebay, G. Mariotti, S. Pedinotti, and M. Savini. A high-order accurate discontinuous finite element method for inviscid and viscous turbomachinery flows. In R. Decuyper and G. Dibelius, editors, *Proceedings of the 2nd European Conference on Turbomachinery Fluid Dynamics and Thermodynamics*, pages 99–108, Antwerpen, Belgium, March 5–7 1997. Technologisch Instituut.
16. F. Bassi and S. Rebay. A high order discontinuous Galerkin method for compressible turbulent flows. In *Discontinuous Galerkin Methods. Theory, Computation and Applications*, volume 11 of *Lecture Notes in Computational Science and Engineering*, pages 77–88. Springer-Verlag, 2000. *First International Symposium on discontinuous Galerkin Methods*, May 24–26, 1999, Newport, RI, USA.
17. F. Bassi and A. Crivellini. A three-dimensional parallel discontinuous Galerkin solver for acoustic propagation studies. *International Journal of Aeroacoustics*, 2:157–173, April 2003.
18. F. Bassi, A. Crivellini, D. A. Di Pietro, and S. Rebay. An artificial compressibility flux for the discontinuous Galerkin solution of the incompressible Navier-Stokes equations. *J. Comput. Phys.*, 218:794–815, 2006.
19. V. Dolejší. Semi-implicit interior penalty discontinuous Galerkin methods for viscous compressible flows. *Commun. Comput. Phys.*, 4:231–274, 2008.
20. G. J. Gassner, F. Lörcher, and C.-D. Munz. A discontinuous Galerkin scheme based on a space-time expansion II. Viscous flow equations in multi dimensions. *J. Sci. Comput.*, 34:260–286, 2008.
21. L. Giraud, Julien Langou, and M. Rozloznik. On the loss of orthogonality in the Gram-Schmidt orthogonalization process. Technical Report No. TR/PA/03/25, CERFACS, 2003.
22. L. Botti. Influence of reference-to-physical frame mappings on approximation properties of discontinuous piecewise polynomial spaces. *J. Sci. Comput.*, 52(3):675–703, 2012.
23. Irene Moulitsas and George Karypis. Multilevel algorithms for generating coarse grids for multigrid methods. In *Supercomputing 2001 Conference Proceedings*, 2001.
24. I. Moulitsas and G. Karypis. MGridGen/ParmGridGen, Serial/Parallel library for generating coarse meshes for multigrid methods. Technical Report Version 1.0, University of Minnesota, Department of Computer Science/Army HPC Research Center, 2001. <http://www-users.cs.umn.edu/~moulitsa/software.html>.
25. F. Bassi, L. Botti, A. Colombo, A. Crivellini, A. Ghidoni, A. Nigro, and S. Rebay. Time Integration in the discontinuous Galerkin Code MIGALE - Unsteady Problems. In Norbert Kroll, Charles Hirsch, Francesco Bassi, Craig Johnston, and Koen Hillewaert, editors, *IDIHOM: Industrialization of High-Order Methods - A Top-Down Approach*, volume 128 of *Notes on Numerical Fluid Mechanics and Multidisciplinary Design*, pages 205–230. Springer International Publishing, 2015.
26. F. Bassi, L. Botti, A. Colombo, A. Ghidoni, and F. Massa. Linearly implicit Rosenbrock-type Runge–Kutta schemes applied to the Discontinuous Galerkin solution of compressible and incompressible unsteady flows. *Comput. Fluids*, 118:305 – 320, 2015.
27. D. N. Arnold, F. Brezzi, B. Cockburn, and D. Marini. Unified analysis of discontinuous Galerkin methods for elliptic problems. *SIAM J. Numer. Anal.*, 39(5):1749–1779, 2002.
28. F. Brezzi, G. Manzini, D. Marini, P. Pietra, and A. Russo. Discontinuous Galerkin approximations for elliptic problems. *Numer. Methods Partial Differential Equations*, 16:365–378, 2000.
29. P. Roe. Characteristic-based schemes for the euler equations. *Annual Review of Fluid Mechanics*, (18):337–365, 1986.
30. A. Harten and Hyman J.M. Self adjusting grid methods for one-dimensional hyperbolic conservation laws. *Journal of Computational Physics*, 50(2):235–269, 1983.
31. P. Tesini. *An h-Multigrid Approach for High-Order discontinuous Galerkin Methods*. PhD thesis, Università degli Studi di Bergamo, January 2008.

32. F. Bassi, L. Botti, A. Colombo, A. Crivellini, N. Franchina, A. Ghidoni, and S. Rebay. Very high-order accurate discontinuous Galerkin computation of transonic turbulent flows on aeronautical configurations. In Norbert Kroll, Heribert Bieler, Herman Deconinck, Vincent Couaillier, Harmen van der Ven, and Kaare Sørensen, editors, *ADIGMA - A European Initiative on the Development of Adaptive Higher-Order Variational Methods for Aerospace Applications*, volume 113 of *Notes on Numerical Fluid Mechanics and Multidisciplinary Design*, pages 25–38. Springer Berlin / Heidelberg, 2010.
33. L. Botti. A choice of forcing terms in inexact Newton iterations with application to pseudo-transient continuation for incompressible fluid flow computations. *Applied Mathematics and Computation*, 266:713 – 737, 2015.
34. M. Wallraff and T. Leicht. Higher order multigrid algorithms for a discontinuous Galerkin RANS solver. *52nd AIAA Aerospace Sciences Meeting - AIAA Science and Technology Forum and Exposition, SciTech 2014*, 2014.
35. M. Wallraff, R. Hartmann, and T. Leicht. Multigrid Solver Algorithm for DG Methods and Applications to Aerodynamic Flows. In Norbert Kroll, Charles Hirsch, Francesco Bassi, Craig Johnston, and Koen Hillewaert, editors, *IDIHOM: Industrialization of High-Order Methods - A Top-Down Approach*, volume 128 of *Notes on Numerical Fluid Mechanics and Multidisciplinary Design*. Springer International Publishing, 2015.
36. A. Ghidoni, A. Colombo, F. Bassi, and S. Rebay. Efficient p -multigrid discontinuous Galerkin solver for complex viscous flows on stretched grids. *Int. J. Numer. Methods Fluids*, 75(2):134–154, 2014.
37. A. Ghidoni, E. Pelizzari, S. Rebay, and V. Selmin. 3d anisotropic unstructured grid generation. *International Journal for Numerical Methods in Fluids*, 51(9-10):1097–1115, 2006.

k	C_L	e	$C_L + e$	η
1	0.131344	+3.98122e-3	0.135325	0.9879
2	0.135374	+5.16825e-4	0.135891	0.9997
3	0.135891	-1.16968e-5	0.135879	1.0633
4	0.135880	—	—	—

Table I. NACA0012 - $M_\infty = 0.5$, $\alpha = 1^\circ$: Values of C_L , for different polynomial degrees, on a mesh of 512 quadrangular elements with quadratic edges, correction e given by the adjoint problem solution and effectiveness of the error estimation $\eta = e/(C_L^{k+1} - C_L^k)$ (where unity would be perfect estimation)

\mathcal{I}	starting mesh (unif.)		last adaptation cycle	
	Mesh (n. el)	\mathcal{E}	Mesh (n. el)	\mathcal{E}
8	3727	1.39912e-2	2512	2.17989e-4
16	1993	9.13139e-2	2062	1.70654e-4
32	1043	5.58066e-2	2274	7.48678e-5
64	538	1.12873e-1	2221	9.04285e-5

Table II. NACA0012 - $M_\infty = 0.5$, $\alpha = 1^\circ$, \mathbb{P}^2 solution: Number of elements and error levels for the starting (uniformly agglomerated) mesh and the last adaptively agglomerated mesh (4th adaptation cycle), \mathcal{R} is made of 26 406 triangular elements

Mesh (n. el)	C_L^{*2}	C_D^{*2}
26 406	0.0191398310	0.0559272138
114 532	0.0190592928	0.0559512300

Table III. NACA0012 - $M_\infty = 0.5$, $\alpha = 1^\circ$, $Re_c = 5\,000$: Reference values of C_L and C_D computed with a \mathbb{P}^2 DG discretizations on the coarse 26 406 and fine 114 532 triangular elements grids